LISTING OF CLAIMS

Claims 1, 5–16, 20, 21 and 24–27 remain as originally filed or as previously presented. No claims are being amended or cancelled by this Response.

1. (Previously Presented) A database cluster which avoids client failure by connecting to multiple nodes of the cluster, the database cluster comprising:

a first computing system including:

a primary connection manager which forms a Transmission Control Protocol/Internet Protocol (TCP/IP) connection with and receives transactions from at least one client, and

a primary database management system (DBMS) which communicates with the primary connection manager to receive the transactions and executes the transactions on data stored in one or more data files; and

a second computing system including:

a secondary connection manager, and

a secondary DBMS which communicates with the secondary connection manager and can access data stored in the one or more data files,

wherein the secondary connection manager is configured to monitor a performance of the first computing system and the primary connection manager is configured to monitor a performance of the second computing system,

and wherein when the secondary connection manager determines that an unbalanced workload exists between the first and the second computing systems, the secondary connection manager transparently assumes the TCP/IP connection, replays incomplete portions of open transactions on the data through the secondary DBMS, and begins to receive additional transactions from the at least one client to be executed against the one or more data files.

2.–4. (Cancelled)

5. (Original) The highly available database cluster of Claim 1, wherein the primary connection manager and the secondary connection manager communicate with one another.

6. (Original) The highly available database cluster of Claim 5, wherein the primary connection manager transmits copies to the secondary connection manager of data packets which include the transactions and responses or acknowledgements to the transactions.

7. (Previously Presented) The highly available database cluster of Claim 5, wherein the primary connection manager and the secondary connection manager exchange statistics in order to monitor the TCP/IP connection.

8. (Original) The highly available database cluster of Claim 7, wherein the statistics include the number of clients connected to the primary connection manager.

9. (Original) The highly available database cluster of Claim 7, wherein the statistics include the number of clients the secondary connection manager can see connected to the primary connection manager.

10. (Original) The highly available database cluster of Claim 7, wherein the statistics include whether the secondary connection manager can communicate with the primary connection manager.

11. (Previously Presented) A primary and at least one secondary connection manager of a database cluster, which manage a connection between at least one client and two or more database management systems (DBMSs), wherein the primary and at least one secondary connection manager can transparently move the connection from the primary connection manager to the at least one secondary connection manager while providing protocols for the connection native to the two or more DBMSs, the primary and secondary connection manager comprising:

a first memory;

a primary connection configured to form a Transmission Control Protocol/Internet Protocol (TCP/IP) connection with a client and to place statements from transactions from the client into the first memory;

a primary protocol shadow configured to retrieve the statements and forward the statements to a primary DBMS;

a secondary memory;

a secondary connection configured to transparently receive transactions from the TCP/IP connection with the client when the secondary connection manager determines that the primary connection has an unbalanced workload and to place new statements from the transactions from the client into the second memory;

at least one process configured to replay any incomplete statements of open transactions; and

a secondary protocol shadow configured to connect to the at least one process until the incomplete statements are forwarded to a secondary DBMS and then to connect to the secondary memory to retrieve the new statements and forward the new statements to the secondary DBMS.

12. (Original) The primary and at least one secondary connection manager of Claim 11, wherein the protocol native to the two or more DBMSs comprises SQL*Net.

13. (Original) The primary and at least one secondary connection manager of Claim 11, wherein the at least one process further comprises:

an import process configured to retrieve the statements from the primary connection and store those statements associated with open transactions; and

a replay process configured to access the stored statements and to forward the stored statements to the secondary protocol shadow.

14. (Original) The primary and at least one secondary connection manager of Claim 11, wherein the secondary protocol shadow is configured to access a log file of the primary DBMS to ensure against replaying of statements of closed transactions.

15. (Original) The primary and at least one secondary connection manager of Claim 11, wherein the primary and secondary connections communicate with one another.

16. (Previously Presented) The primary and at least one secondary connection manager of Claim 15, wherein the primary connection and the secondary connection exchange statistics in order to monitor the TCP/IP connection.

17.–19. (Cancelled)

20. (Previously Presented) A method of providing transparent fail-over to a client connection when a primary database management system (DBMS) has an unbalanced workload, the method comprising:

monitoring statistics of a client Transmission Control Protocol/Internet Protocol (TCP/IP) connection between a first DBMS and a client, wherein said monitoring is performed by a connection manager of a second DBMS;

determining from the statistics a need to transparently move the client TCP/IP connection to the second DBMS while keeping the client TCP/IP connection alive from a perspective of the client;

rerouting the TCP/IP client connection to the second DBMS;

replaying any statements from open transactions rolled back when the client TCP/IP connection was moved from the first DBMS; and

establishing communication between the second DBMS and the client over the client TCP/IP connection.

21. (Previously Presented) A data processing system which provides transparent fail-over to a client connection, the data processing system comprising:

a first host configured to accept a Transmission Control Protocol/Internet Protocol (TCP/IP) connection from a client;

a connection manager of a second host which reroutes the Transmission Control Protocol/Internet Protocol (TCP/IP) connection to the second host without recognition by the client when the connection manager determines that the first host has an unbalanced workload; and

a replay process which forwards to the second host at least one incomplete statement from open transactions when the Transmission Control Protocol/Internet Protocol (TCP/IP) connection was moved from the first host, wherein the connection manager establishes communication between the

second host and the client over the Transmission Control Protocol/Internet Protocol (TCP/IP) connection.

22.–23. (Cancelled).

24. (Previously Presented) A database cluster for moving client connections with one or more client program applications from a first host to a second host, the database cluster comprising:

a first database management system (DBMS) configured to access data stored in one or more data files;

a first connection manager configured to form a Transmission Control Protocol/Internet Protocol (TCP/IP) connection with a client computer, wherein the first connection manager is further configured receive the transactions from the client computer and to communicate the transactions to the first DBMS;

a second DBMS configured to access the data stored in the one or more data files; and

a second connection manager configured to monitor the TCP/IP connection between the first connection manager and the client computer, the second connection manager further configured to transparently assume the TCP/IP connection with the client computer while keeping the client TCP/IP connection alive from a perspective of the client.

25. (Previously Presented) The database cluster of Claim 24, wherein the second connection manager is configured to transparently assume the TCP/IP connection with the client computer when the second connection manager detects a failure of the TCP/IP connection between the first connection manager and the client computer.

26. (Previously Presented) The database cluster of Claim 24, wherein the second connection manager is configured to transparently assume the TCP/IP connection with the client computer when the second connection manager detects a failure of the first DBMS.

27. (Previously Presented) The database cluster of Claim 24, wherein the second connection manager is configured to transparently assume the TCP/IP

connection with the client computer when the second connection manager detects an unbalanced workload of the first DBMS.